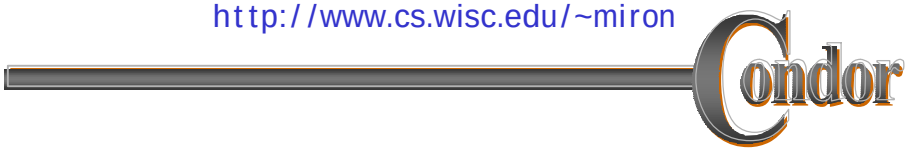
Request Execution: a Tier 3 Perspective

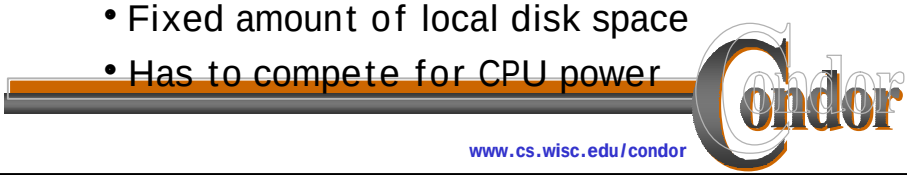
Miron Livny
Computer Sciences Department
University of Wisconsin-Madison
miron@cs.wisc.edu
http://www.cs.wisc.edu/~miron



Request

Use local resources (processing and storage) to apply program P to a subset of files from collection C that meets constraint Q

- Files must be staged locally
- No output
- Fixed amount of local disk space
- Has to compete for CPU power



Expectations

- No need to know location of C
- Reliable
- Error Reporting
- Efficient
- Progress Reporting

www.cs.wisc.edu/condor

Method

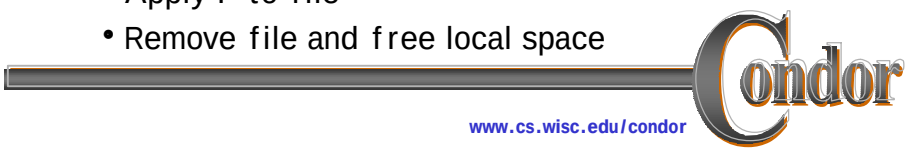
Develop a Request Executor (ReqEx) and test its end-to-end functionality in a testbed with the following components:

- Query Estimator (QE) at LBL
- Meta Catalog (MCAT) provided by SRB
- Collection replicated at PPDG sites (LBL,SDSC, ...)
- File staging via SRB
- Condor pool and disk cache in Wisconsin

www.cs.wisc.edu/condor

Execution Steps

- Receive C and Q from user
- Locate QE for C
- Send Q to QE and get list of files
- Obtain file replication information from MCAT
- For each file
  - Select storage location
  - Allocate local space and stage file
  - Apply F to file
  - Remove file and free local space



Building the Testbed

- Install SRB and QE related software in Wisconsin
- Obtain SRB account
- Install SRB software in LBL
- Register LBL PPDG collection in MCAT
- Replicate (and register) PPDG collection at SDSC.
- Establish disk cache at Wisconsin

ReqEx prototype

- Second version built with software components from Condor
- Prototype works
- Only uses the Sget command of SRB
- Does not support replication
- QE location hard coded.

Main Lessons

- Installation of software components can not be done by “the locals”
- Lack of support for inter-component exception handling a major obstacle
- Need for a “development” testbed and a “production” testbed
- Software upgrades a problem.